オブジェクト指向分散環境 OZのプロセス内セキュリティ

1K－7

濱崎 陽一
電子技術総合研究所
hamazaki@etl.go.jp

西岡 利博
三菱総合研究所
nishioka@mri.co.jp

塚本 享治
電子技術総合研究所
tukamoto@etl.go.jp

1　はじめに

著者らが開発中のOZは他のプログラマが開発したプログラムの再利用により、アプリケーションの開発を容易にするのみならず、他のプログラマが開発したネットワーク上のオブジェクトへもメソッド起動でき、また相互運用性を保ったままで独立にクラスの拡張が可能なオブジェクト指向分散環境である。そのためインタフェースだけ同じで実装が異なるようなオブジェクトが実行時に導入されても実行できるように、ネットワークから必要なクラスが配送される機構を備えている。

こうしたシステムでは不特定多数のユーザから未知のクラスのオブジェクトを受けとり、それを実行する際の予期しない振舞いに対して安全性を保証するセキュリティ機能が重要であり、OZではユーザ認証に基づくセキュリティと、実行プロセス内の挙動に対する低レベルのセキュリティの二段階で実現する。本稿では後者のプロセス内セキュリティについて述べる。

2　OZのセキュリティ

OZはネットワークを介してアクセス可能なグローバルオブジェクトとグローバルオブジェクトを構成する部品となるローカルオブジェクトの2種類のオブジェクトから成るモデルを採用している。ひとつのグローバルオブジェクトとそれを構成するローカルオブジェクトを合わせてセルと呼ぶ。ローカルオブジェクトはそれが属するセルの外部からアクセスできないので、他のグローバルオブジェクトへのメソッド起動の引数あるいは返り値となる場合には、それが参照しているローカルオブジェクトも含めて構造を保ったままコピーして引き渡される。未知のクラスのオブジェクトを受けとった場合にも、クラスの配送を受けることにより、実行できる。(以下、ローカルオブジェクトを単にオブジェクトと記す)

メソッド起動の連鎖によって生成されるスレッドの連鎖をOZのプロセス(以下、曖昧でない箇所では単にプロセスと呼ぶ)と呼び、グローバルオブジェクトのメソッド起動に関連するプロセスによって望ましくない操作が行われないようにするのがここで述べるプロセス内セキュリティである。

プロセス内セキュリティとして次の2種類を考える。

OZ以外のシステムをOZから守る OZを導入した事によって予期しないファイルの破壊／窃取や他のシステムのプロセスの破壊を防ぐものである。ファイル、ソケット、サブプロセスなどの操作によるものが考えられる。

OZのオブジェクトを守る OZのオブジェクトに蓄えられた重要な情報の破壊／窃取を防ぐものである。そのオブジェクトに対する直接のメソッド起動による破壊／窃取や、他のグローバルオブジェクトとのメソッドに伴うクローニングによる窃取が考えられる。

プロセス内セキュリティはプロセスの個々の操作に対して課せられるセキュリティであるから、その実現には柔軟性よりも高速性を優先し、高速性の期待できないユーザ認証を行わず、危険でない／危険である可能性があるの2種類の認証に留める。危険でないものを緑、危険である可能性があるものを赤と呼ぶ。

外部から到来したオブジェクトは予期せぬ動作をする可能性があり、赤である。また赤オブジェクトが命じた操作は危険性を伴うものとして扱う必要があるが、呼出の連鎖を遡ってその操作を命じたオブジェクトに赤が含まれていたかどうかを判断するのは困難であるので、プロセスにも色をつけた。オブジェクトとプロセスの色は、表1のようになる。

プログラミングを容易にするために、重要な情報を持つオブジェクトなどを指定する事により保護するのではなく、特に許可しない限り一切危険性のある操作をさせないスタイルを取った。

3　プロセス内セキュリティの実装

OZはJavaを用いて実装をすすめており、OZ言語で開発されたクラスはOzObjectというクラスを継承したJavaのクラスにコンパイルされる。

プロセス内セキュリティ機能の大半は、OzObjectのメソッドとして提供するサービスとコンパイラによるコード生成により実現する。OzObjectのサービスには、(a1)オブジェクトの色を知る、(a2)自プロセスの色を知る、(b1)オブジェクトの色を変える、(b2)プ

A Security mechanism within a process in OZ: An Object-Oriented Distributed Systems Environment
Yoichi Hamazaki(Electrotechnical Laboratory),
Toshihiro Nishioka(Mitsubishi Research Institute, Inc.),
and Michiharu Tsukamoto(Electrotechnical Laboratory)

Java is a trademark of Sun Microsystems Inc.

表 1: オブジェクトとプロセスの色

オブジェクトの色	他のグローバルオブジェクトから到来したオブジェクトは赤
プロセスの色	赤オブジェクトのメソッドを実行中のプロセスはそのセル内では赤。
オブジェクトの属性変更	緑プロセスはオブジェクトの色を変更できる。
プロセスの属性変更	緑プロセスは、赤オブジェクトのメソッドを実行しようとすると赤になるが、 そのメソッドの実行が終了すると緑に戻る。 緑オブジェクトは、それを実行中のプロセスを赤にできる
オブジェクトの生成	赤プロセスが生成するオブジェクトは赤。
メソッド起動	赤プロセスは緑オブジェクトのメソッドを起動できない。
プロセスのフォーク	赤プロセスがフォークすると赤プロセスが生成される。
オブジェクトのクローニング	赤プロセスは緑オブジェクトをクローニングできない。

```
OzMethod1() {
  /* c2 */
  checkSecureInvocation();
    ...
  /* メソッド起動の部分 (c1) */
  if(changeProcessRedIfNecessary(callee)){
    /* プロセスは一時的に赤 */
    try {
         callee.invoke ();
        } finally {
         changeProcessColor (GREEN);
        }
  } else { /* プロセスとオブジェクトは同色 */
    callee.invoke ();
  }
    ....
}
```

図 1: コンパイラの出力コード例

ロセスの色を変える、(c1) 呼び出すメソッドが赤オブジェクトのものである場合にはプロセスを赤に変える、(c2) 赤プロセスによる緑オブジェクトのメソッド起動を検出するの六つがある。

Java 言語のパッケージによるアクセス保護機構を利用して、一般のプログラムはこれらのサービス以外の方法では色を変えることができないようにしている。b1,b2 のランタイムは赤プロセスから呼ばれると例外を発生する。また、OZのプロセスでない Java のスレッドは赤プロセスとして扱う。

赤オブジェクトのメソッド起動時にプロセスを赤に変え (c1)、その延長上での緑オブジェクトのメソッド起動を検出 (c2) することにより赤オブジェクトが命じた操作による緑オブジェクトのメソッド起動を禁止している。メソッド起動のコードはコンパイラにより図 1のように出力される。

クローニングには Java RMI の Serializer[2] を用いており、OzObject の writeObject メソッドで赤プロセスによる緑オブジェクトのコピーを禁止している。また、OZ以外のシステムの保護については、Java と同様に Java の SecurityManage をカスタマイズすることにより行うが、その際プロセスの色に応じて対応を変えるなど、設計に幅をもたせることができる。

また、配送されるクラスとしてはOZ言語で開発されたクラスだけでなく Java のクラスも許すが、プロセス内セキュリティ機能を無効にする可能性のある以下のメソッドが定義されない事が検査される。1) ネイティブメソッド 2)writeObject/readObject メソッド (オブジェクトのクローニングの意味を変更する可能性のあるメソッド) 。

試験的に実装したコードでメソッド起動のオーバヘッドを JDK 1.0.2 を用い、Solaris 2.5.1(Ultra-1) および Windows NT(PentiumPro 200MHz) で測定した結果 (比率) を表 2に示す。

表 2: メソッド起動の性能

プロセス	callee	Solaris2.5.1	Windows NT
Java	Java	1	1
緑	緑	4.2	4.2
緑	赤	7.1	7.3
赤	赤	4.8	4.8

4 まとめ

ネットワーク上のソフトウェアを積極的に再利用することにより分散アプリケーションの開発を容易にするオブジェクト指向分散環境OZでは、未知のクラスおよびそのオブジェクトを受け入れて実行することが必要となるため、セキュリティ機能は重要な意味を持つ。

ここでは、低レベルのセキュリティを実現するプロセス内セキュリティについて述べたが、ユーザ認証に基づく柔軟な高レベルのセキュリティの設計も現在進めており、両者によってワールドワイドな環境でも安全に利用できるセキュリティを実現する予定である。

本研究は、情報処理振興事業協会 (IPA) の「創造的ソフトウェア育成事業」の一環として行われたものである。

参考文献

[1] The Java Language Specification,Ver.1.0, Sun Microsystems Inc.,Aug. 1996

[2] Java Object Serialization Specification ,Rev.0.9, Sun Microsystems Inc.,May 1996